Attorney Docket: 80398.P198 *PATENT*

UNITED STATES PATENT APPLICATION

FOR

METHOD AND APPARATUS FOR ERROR DATA RECOVERY

INVENTORS:

TETSUJIRO KONDO

SUGATA GHOSAL

YASUHIRO FUJIMORI

JAMES J. CARRIG

YASUAKI TAKAHASHI

TAKAHIRO NAGANO

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026

(408) 720-8598

"Express Mail" mailing label number EL236230522US

Date of Deposit Feb 12, 1999

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231

Cristina M. Kennard
(Typed or printed name of person mailing paper or fee)

Cristina M Kennard 2/12/99
(Signature of person mailing paper or fee) *Date*

METHOD AND APPARATUS FOR ERROR DATA RECOVERY

BACKGROUND OF THE INVENTION

1. FIELD OF THE INVENTION

The present invention relates to the recovery of lost or damaged encoded data. More particularly, the present invention relates to the minimizing of quality degradation caused by error propagation in bitstreams containing lost or damaged encoded data.

2. ART BACKGROUND

It is quite common to compress data to minimize transmission or storage bandwidth requirements. One type of compression process is referred to as variable length coding. In variable length coding processes, the signal is typically divided into several localized regions, also referred to as blocks, and is coded by quantizing each region according to its signal activity level. In an exemplary signal such as signals representative of digital images, different parts of the images have different activity levels and are therefore coded with a different number of quantization bits.

The information regarding the number of quantization bits used for coding different regions is used by the decoder to delineate the respective quantization bits of each block from the received bitstream, which in turn is used to decode the blocks. Therefore, when the information regarding the number of quantization bits used to encode data is lost, a recovery process is implemented to estimate the number of quantization bits used to generate the codes representative of the data. If the number of quantization bits is not accurately estimated, the error incurred will propagate through the bitstream as

the decoder is unable to determine the location of the end of one block, and therefore the beginning of the next block.

One type of variable length encoding is known as Adaptive Dynamic Range Coding (ADRC). For further information regarding ADRC, see, "Adaptive Dynamic Range Coding Scheme for Future HDTV Digital VTR", Kondo, Fujimori, Nakaya, Fourth International Workshop on HDTV and Beyond, September 4-6, 1991, Turin, Italy.

In one example of ADRC, blocks are encoded using a minimum pixel value (MIN), the dynamic range (DR) of pixel values in the block, the motion flag (MF) indicative of temporal activity and the quantization codes (Q codes) representative of each pixel in a block. The Q codes are generated based upon a minimum value, quantization step and the original pixel value. The quantization step is determined by the DR of the block and the number of quantization bits (Qbit), wherein the Qbit is a function of DR.

When the encoded bits are received in the decoder, the Qbit and MF information of each block is needed to delineate the portion of the bitstream corresponding to each block , and in turn to decode the block. If the DR and/or MF of the block is damaged, it is necessary to recover or estimate the information in order that the blocks can be decoded. At the same time, in case of a recovery failure, this error will likely result in incorrect decoding of the rest of the blocks and, in turn, severe picture degradation, since the starting point of subsequent blocks in the bitstream will be incorrectly identified.

Ins a1

SUMMARY OF THE INVENTION

The present invention provides a mechanism for preventing quality degradation of decoded data during the decoding of encoded data. In one embodiment, error propagation is detected and corresponding data is flagged. An error recovery process is then applied to the flagged data. In an alternate embodiment, candidate hypotheses are calculated for lost/damaged data. A score distribution is used for detection of the false candidate hypotheses. The data are flagged if their score distribution is within a range defined by a threshold and an error recovery process is applied to recover those data having associated error flags set.

BRIEF DESCRIPTION OF THE DRAWINGS

The objects, features and advantages of the present invention will be apparent to one skilled in the art in light of the following description in which:

Figure 1a illustrates one embodiment of a system of the present invention, **Figure 1b** illustrates an alternate embodiment of the system of the present invention, and **Figure 1c** illustrates an alternate embodiment of the system of the present invention.

Figure 2 is a block diagram illustration of one embodiment of a sub-system for the recovery of block parameters.

Figure 3 is a block diagram illustration of one embodiment of a sub-system for determining a measure for different hypotheses.

Figure 4 is a block diagram illustration of one embodiment of a sub-system for scoring the different hypotheses.

Figure 5 illustrates an example of the effect of error propagation on score distribution.

Figure 6 shows exemplary histograms of score distribution for recovery of block parameters.

Figure 7 illustrates an example of successive score distribution following Qbit and Motion Flag recovery.

Figure 8 is a block diagram illustration of one embodiment of a sub-system for detection of error propagation.

Figure 9 is a flow diagram illustrating one embodiment of a process for recovering lost or damaged data.

Figure 10 is a flow diagram illustrating an alternate embodiment of a process for recovering lost or damaged data.

Figure 11 is a flow chart illustrating one embodiment of a pixel error recovery process.

Figure 12 illustrates parameters of an audio signal.

DETAILED DESCRIPTION

In the following description, for purposes of explanation, numerous details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required in order to practice the present invention. In other instances, well known electrical structures and circuits are shown in block diagram form in order not to obscure the present invention unnecessarily.

The system and method of the present invention provides an innovative mechanism for preventing quality degradation caused by error propagation in variable length encoded data. The discussion herein is directed to the recovery of image data, and in particular, Adaptive Dynamic Range Coding (ADRC). However, the invention is not limited to image data and can be applied to other types of correlated data including audio data. Furthermore, the present invention is not limited to ADRC; other variable length encoding processes may also be used. For example, the present invention is applicable to coding processes using Discrete Cosine Transform (DCT). In one embodiment, DCT coefficients of individual blocks can be quantized based upon respective activity levels, and the quantized coefficients are transmitted.

Ins a2

~~One embodiment of a system that operates in accordance with the teachings of the present invention is illustrated in **Figure 1a**. The encoded bitstream of data, error flags indicating portions of data that are lost or damaged (lost/damaged), and decoded data (referred to in this example as image data) are input to Qbit and Motion Flag recovery circuit 10. The Qbit and Motion Flag recovery circuit 10 generates hypotheses, referred to herein as candidate hypotheses, of possible number of quantization bits (Qbit) used and motion flag (MF) values. As will be described in more detail below, a score is generated with respect to each hypothesis and the hypothesis with the best~~

score, e.g., minimum score, is selected as the recovered values. The data decoded using the selected hypothesis is output as the decoded data to the memory 30.

a2 con't

The scores and error flags corresponding to data decoded using the selected hypothesis is input to error propagation detection circuit 20. As will be discussed below, error propagation detection circuit 20 evaluates candidate hypotheses to generate candidate hypotheses results used to detect error propagation. In one embodiment, circuit 20 examines the score distribution and detects error propagation due to false candidate decoding. Other evaluation techniques may be used, including evaluation of score distribution patterns or other metrics. Pixel error recovery block 40, receives the decoded data error flags as well as flags that may have been generated by the error propagation detection circuit 20 to indicate that pixel error recovery is warranted and performs a pixel error recovery process to recover pixel data that may not be correct due to error propagation.

An alternate embodiment is illustrated in **Figure 1b**. In this embodiment, a general purpose or specially configured processing system 50 performs the methods described herein. In this embodiment, the processing system 50 includes a processor 55, a memory 60 and input/output circuitry 65. Other circuitry not shown herein may also be included.

In one embodiment, the memory can store instructions, which when executed by processing system 50, perform the methods described herein. Alternately, the instructions may be stored on other storage media or transmitted across a transmission media, such as a network, to the processor 55. Memory 60 can also be configured to store data used and generated as described herein.

Input/output circuitry 65, receives the encoded data, error flags and image data for processing by processing system 50 and outputs the recovered data generated.

Ins Q3

An alternate embodiment is illustrated by **Figure 1c**. In this embodiment, the bitstream of data is input into decoder 70 and candidate decoder 72. The decoder 70 decodes those portions of the bitstream that are decodable using standard available decoding processes, i.e., portions containing no errors. Candidate decoder 72 generates candidate decodings and selects a best candidate decoding as discussed herein. Error propagation detection circuit 74 detects errors which occur due to error propagation, for example, those errors caused by selection of an incorrect candidate decoding. The prevent degradation circuit 76 receives the decoded data output by decoder 70 and error propagation flags generated by error propagation detection circuit 74 and performs block processing to recover blocks of data flagged to have errors. In one embodiment, block processing estimates block data using neighboring block data. Other block processing techniques may be used.

Pixel error flag circuit 80 receives flags indicating the pixel data which may contain errors. Typically, the flags are generated using known Error Correction Code (ECC) techniques.

Pixel error recovery circuit 78 receives input from prevent degradation circuit 76 and pixel error flag circuit 80 and performs a pixel error recovery process data for having corresponding error flags set. A pixel error recovery process, such as the classified adaptive error recovery process discussed below, may be used.

A more detailed block diagram of one embodiment of the Qbit and Motion Flag recovery circuit is shown in **Figure 2**. In this embodiment, candidate decoding and error measure circuit 205 determines possible

hypotheses, i.e., values of Qbit and/or MF, and generates candidate decodings of Q codes based upon each of the hypotheses. Alternately, the hypothesis can be evaluated in the encoded domain, eliminating the need for decoding the data for all hypotheses.

An error measure is generated for each hypothesis. In one embodiment, an error measure may be determined based upon how well the candidate decoding fits into with other decoded data. For example, the measure may show how well correlated the candidate decoding is with other decoded data. Linear error, square error and Laplacian measurements may also be used.

One measure that can be used is square error measurements on decoded data. In one embodiment, the decoded domain square error measurement can be obtained using the following formula:

$$SquareError(HypNo) = \sum_i \sum_{j \in R(i)} \left[y_i' - y_j\right]^2$$

where y_i' represents the i-th decoded pixel value of the block being recovered for the hypothesis HypNo. and y_j represents one of the neighboring decoded pixel values of the i-th pixel, $R(i)$ represents a set of neighboring pixels to the i-th pixel and

$$y_i' = MIN + \frac{DR(q_i + 0.5)}{2^Q}$$

where DR is the recovered dynamic range value, q_i is the i-th Q code and Q is the Qbit number corresponding to the hypothesis HypNo.

In one embodiment, the data in the encoded domain is evaluated with respect to each hypothesis to generate corresponding error measures. For

example, a linear error measure may be determined from encoded data as follows:

$$LinearError(HypNo) = \sum_i \sum_{j \in R(i)} \left| a1_i - a2_j \right|$$
$$a1_i = adj(q1_i)$$
$$adj(q1_i) = 2^{5-Q1} q1_i + 2^{4-Q1}$$
$$a2_j = adj(q2_j)$$
$$adj(q2_i) = 2^{5-Q2} q2_j + 2^{4-Q2} + Offset$$
$$Offset = \frac{1}{N} \sum_i \sum_{j \in R(i)} (q1_i - q2_j)$$

where $q1_i$ represents the Q code of the i-th pixel of the block being recovered (block 1), $q2_j$ represents the Q code of a neighboring pixel of the i-th pixel of a neighboring block (block 2), N is the number of neighboring-pair relations, Q1 and Q2 respectively represent the hypothesized Qbit number of the block being recovered and the Qbit number of the neighboring block. R(*i*) represents the neighboring pixels with respect to the i-th pixel of the block being recovered, and a1i and a2j represent a rescaled value of a2j. The function adj() as defined above performs a Q code rescaling process to normalize the values for more accurate measurements.

Figure 3 illustrates one embodiment of a circuit which implements the encoded domain linear error measure. Candidate decodings 305, 307, 309, 311, 313, 315 are performed based upon the different hypothesis, e.g., hypotheses 0-5. For each hypothesis, the hypothesized Q code for the block to be recovered (q1), the Q code for a neighboring block (q2) and Q code error flags determined for the particular hypothesis are input to a linear error measurement circuit 317, 319, 321, 323, 325, 327 to generate the linear error measure, Linear Error 0-5.

The measure may be determined in accordance with the equations described above.

Ins a4

Referring back to **Figure 2**, the error measures are input to scoring circuit 210 which determines which hypothesis generates the best score. A variety of techniques can be used to determine the best score. For example, confidence weighted scoring, majority based scoring, simple accumulation based scoring and majority decision scoring can be used. In one embodiment the score is determined as follows:

$$score(i) = \sum_j G\left(\frac{m(i,j) - \min(j)}{m(i,j) + \min(j)}\right)$$

where m(i, j) represents the j-th measurement for hypothesis i for the block or block group (e.g., 3 blocks which form a group), min(j) is the minimum of the j-th measurement among different hypotheses, and G() is an identity function or a monotonically increasing function, depending on the application. G may be selected to be a function that increases the sensitivity of the scoring such that incorrect hypotheses will be clearly distinguishable from the correct hypothesis. For example, a square or linear function may be used. Alternately, G may be a constant, including one having a value of unity.

Alternately, if an accumulation function of, for example, linear error measures, is performed to produce a score, the score would be determined as follows:

$$score(i) = \sum_j m(i,j)$$

In this example if the measure is linear error, the hypothesis with the lowest score is determined as the best score. However, it should be realized that the optimal choice can depend on the type of measure, application and compression algorithm used.

Ins a5

~~**Figure 4** illustrates one embodiment of a circuit to determine the best score. In this embodiment, the error measures, e.g., linear error 0-5 are respectively input to function G[·]0 to G[·]5 405, 407, 409, 411, 413, 415. The minimum score can also be input to functions G[·]0 to G[·]5 and the values output are summed by adder logic 421, 423, 425, 427, 429, 431 to produce scores Score(0) to Score(5). In this embodiment, the minimum value determined by minimum circuit 435 is identified as the best score.~~

~~Referring back to **Figure 2**, the best score is input to the selectors 215 and 220 which respectively output decoded data of the selected hypotheses corresponding to the best score and error flags identified with particular portions of the decoded data decoded according to the selected hypothesis.~~

As noted earlier, if the hypothesis chosen is not correct, subsequent blocks of data will also be incorrectly decoded. This is illustrated with respect to **Figure 5**. Representation 505 illustrates an example of encoding blocks in signal space. For example, representation 505 may be a representation of an image frame which contains blocks of data.

In one embodiment, image 505 shows a plurality of pixels, e.g., 510, 511, 514, 516, 518 which are encoded to produce an encoded bitstream 520. If, for example, block *i* is incorrectly decoded such that the length of the Q codes is not accurate, the error will propagate to subsequent blocks, e.g., blocks i+1, i+2, etc., as the starting points of the subsequent blocks are not correctly identified. As an illustration, bitstream 520 illustrates the correct start point 550 for block *i* and incorrect start point 555 for block i+1.

Ins a6

Whenever the selected hypothesis is wrong, scores are very similar among candidate hypotheses. The threshold point can be selected such that all incorrect recovery results are surely detected. Typically, such a threshold also results in a few false alarms, i.e., some correct recovery results are detected as incorrect (shown as the shaded area in **Figure 6**), and a subsequent error recovery process such as the pixel recovery process below, is used to recover the corresponding pixels. **Figure 6** further illustrates a proper threshold selection.

The thresholds can be empirically determined for a particular application using known data. For example, for 4 bit ADRC encoding, the threshold may be set to approximately a value of 120.

Ins a7

In one embodiment, score distributions in successive recoveries are used to reinforce the error detection decision. Given that the current hypothesis is correct, the recovery start point of a next block or group of blocks in the bitstream is also correct. If the start point is correct, candidate hypotheses for subsequent blocks or groups of blocks will generate pixels that exhibit highly correlated properties with respect to neighboring blocks resulting in large score distributions. If the start point is incorrect, candidate hypotheses for subsequent blocks or group of blocks will generate pixels that are uncorrelated with the neighboring pixels resulting in uncorrelated score distributions.

The score distributions previously computed can therefore be used to flag propagation errors. For example, referring to **Figure 5**, score distribution 550 illustrates the selection of an incorrect hypothesis. As noted above, if an incorrect hypothesis is chosen, the starting points for subsequent blocks may not be correct. Score distributions for subsequent blocks can be used to determine error propagation. For example, score distribution 555 shows the score distribution for block i+1 when the starting point is incorrect and score

distribution 560 shows the score distribution for block i+1 when the starting point is correct.

Ins A8

The likelihood that the hypothesis with the best score is correct may be indicated by the score distribution *sd(i)* among candidate hypotheses. The score distribution can be measured in terms of statistics of various order, including standard deviation, average, median, difference of best score and second best score, difference of best score and average of best scores, etc. A score distribution may be chosen so that the score distribution curve for the correct recovery is completely non-overlapping with the score distribution curve for incorrect recovery.

A measurement criterion can be chosen that is highly sensitive to scoring variation among candidate hypothesis. The optimal choice depends on the measurement and scoring techniques used in conjunction with the parameter, i.e., Qbit and Motion Flag, recovery technique as well as the type of data, for example, audio or video.

For example, if linear error measurements are used in conjunction with simple accumulation based scoring of the linear error measurements, the difference between the best and second best linear error scores may be used as a metric for score distribution. Thus, continuing with the present example, the score distribution for the i-th block or group can be measured as sd(i) = score(second best) - score(best), where score(best) represents the score for the contemplated best hypothesis, and score(second best) represents the score for the contemplated second best hypothesis.

The likelihood that a chosen hypothesis is correct may also be determined based on a compatibility measurement. One embodiment of this concept may utilize the decoded domain square error measurement. Pixels belonging to a localized block are highly correlated with neighboring pixels,

and as a result, the square error measurement may yield a low value, although in general, the higher the dynamic range, the higher the square error. However, for a given dynamic range, an incorrect hypothesis results in a much higher square error measurement than that resulting from a correct hypothesis. Thus, the ratio of square error measurement and dynamic range can be alternatively used in place of the score distribution measurement sd.

As noted earlier, the error detection process based on score distribution measurements sd(*i*) for a block or group *i* can be reinforced by combining the score distribution measurements of the successive blocks in the bitstream, e.g., i + 1, i + 2...i + W, where W is a length of what is referred to herein as reinforcement window. In one embodiment when groups of 3 blocks are processed together, W may be equal to 2. Other values of W may be selected depending upon the application and performance desired.

Figure 7 shows a typical score distribution sd of successive blocks or groups (referred to herein collectively as block units) following recovery for the i-th block unit. If the current recovery result for block *i* is incorrect, recovery start points of all successive blocks (i + 1...i + W) within the reinforcement window W becomes incorrect and as a consequence, the corresponding score distribution measurements are small. If the current recovery result is correct, score distribution measurements of successive block units are large. Therefore, score distributions or successive units within a predetermined reinforcement window can be combined to generate an integrated robust measurement for error propagation detection.

Several schemes can be employed for combining successive score distribution measurements, including empirically weighted averaging within the reinforcement window, or simple addition of score distribution measurements within the window.

For example, if the combined score for unit i is:

$$comb\_sd(i) = \sum_{j=0}^{W} sd(i+j) \leq comb\_thr,$$

the recovery result is detected as an incorrect one if the current score sd(i) is less than or equal to a threshold value ind-thr and comb_thr is a preset threshold value for multiple block units within the reinforcement window.

As noted earlier, the threshold values may be empirically determined. For example, the individual threshold may be a value of approximately 120, and the combined threshold may be a value of approximately 800 when W=2.

Alternately, a majority decision can be employed. For example, if:

$$comb\_sd(i) = \sum_{j=0}^{W} \min(1, \max(0, sd(i+j) - \tau(j)))$$

is less than W/2, the recovery result is marked as an incorrect one. $\tau(j)$ is a constant or variable threshold within the reinforcement window. In one embodiment, $\tau(j)$ is a predetermined constant that increases in value according to the block number in the block sequence being examined. For example, for 3-block units, $\tau(0)$ may be approximately equal to 120, $\tau(1)$ may be approximately equal to 200 and $\tau(2)$ may be approximately equal to 300.

Other methods of combining the information of a plurality of blocks or groups of blocks can also be used.

Figure 8 is one embodiment of a circuit for error propagation estimation. The scores for a particular block are evaluated to determine the best and second best score. This is determined by circuits 805 and 810. The difference between the best and second best score is determined by element 815. The scores are then temporarily stored in registers 820, 825, so as to accumulate scores in the present embodiment for a group of three blocks within a reinforcement

window of length 3. The difference values are then summed to perform by adder 830 to produce a combined value which is compared to the combined threshold 840 by comparator 845. If the threshold is not exceeded, an error flag is raised which is then combined by OR gate logic 850 with other error flags which may have been previously determined to generate the pixel error flags that indicate that a pixel recovery process is required for these particular pixels.

Ins a9

Figure 9 illustrates one embodiment of a process for recovering data, and in particular for detecting error propagation in a bitstream of data. At step 905 the start point of a block or group of blocks is identified. If an error is found in the bitstream corresponding to that start point, a recovery process is initiated to provide candidate decodings. For example, in one embodiment candidate decodings of Motion Flag and Qbit data may be generated based upon specified parameters. At step 920, each hypothesis is scored and the candidate with the best score is selected, step 925. The score distribution sd, is measured.

At step 935, if the error flag, DetectErrorFlag, is set, then the score distribution is combined with a prior combined score distribution variable comb_sd, and a counter (j) is incremented, step 950. Thus the combined score distribution comb_sd is generated for a predetermined number of blocks within the reinforcement window W.

The combined score distribution value is then compared to a combined threshold value comb_thr to determine whether subsequent blocks require pixel recovery as the error in a previous block has caused error propagation to occur in the subsequent blocks. At step 940, if an individual score distribution is within a range defined by the threshold value, e.g., does not exceed the individual threshold value, the error flag is set, step 945. Further, a sum of subsequent score distributions is generated and a counter maintained to sum

up the score distributions for W blocks or groups of blocks within the reinforcement window, step 955.

Once the combined score distribution has been generated for the number of blocks within the reinforcement window, at step 960, the combined score distribution is compared to the combined threshold value comb_thr. If a combined score distribution is not within the range defined by the threshold value, e.g., exceeds or equals the combined threshold value, the error flag and counter are reset and the next block is examined, steps 965, 905. This is indicative of the fact that error propagation did not occur and pixel recovery is not required for those blocks. However, if the combined score distribution is within the range, e.g., less than the combined threshold, step 960, error propagation flags are set for remaining undecoded pixels, step 970. The error propagation flags are optionally combined with other error flags, step 974, and pixel error recovery processing is performed on the error flagged pixels, step 978.

One embodiment of the process is illustrated in **Figure 10**. **Figure 10** performs steps 1005, 1010, 1015, 1020, 1025, 1030, 1035, 1040, 1045, 1050, 1055, 1060, 1065, that are similar to those steps discussed above with respect to **Figure 9**. In addition, the process as set forth in **Figure 10** marks the past W blocks for pixel recovery if the combined score distribution is within the range defined by the combined threshold value, step 1070. If there are more blocks to process, the error flag and the counter are reset and the next block examined, step 1075 and 1005. Pursuant to steps 1075 and 1080, pixel error recovery is performed on those pixels with associated error flags set, step 1075, is optional. Thus, in one embodiment a pixel recovery process is performed on pixel errors flagged due to error propagation.

The embodiment set forth in **Figure 10** allows for those instances where the start points may naturally realign even though earlier blocks contained erroneous start points. Thus, pixel recovery will only be applied to those blocks or groups of blocks that have been examined and determined in the range defined by the threshold.

As noted above, once the error flags are set, whether due to error propagation or earlier error detection, a pixel recovery process is performed (e.g., steps 978 **Figure 9**, step 1080 **Figure 10**). In one embodiment, neighboring blocks are used to recover pixels or pixel data contained in erroneous blocks. For example, classified adaptive error recovery processing can be used to generate the pixels using available neighboring pixels. This is disclosed in U.S. Patent No. 5,469,216, which is incorporated herein by reference. Alternately, a classified adaptive error recovery process such as is explained below may be used.

Classification with respect to a deteriorated input signal, e.g., a signal containing lost/damaged data, is performed according to the input signal characteristics. The correct adaptive filter is prepared for each class prior to error recovery processing. More than one classification method may optionally be used to create the plurality of classes. Created classes may include a motion class, an error class, a spatial activity class, or a spatial class.

Classified adaptive error recovery is the technology which utilizes classified adaptive filter processing. A correct classification with respect to the deteriorated input signal is performed according to the input signal characteristics. An adaptive filter is prepared for each class prior to error recovery processing. A plurality of classes is generated based upon characteristics of the data points. The data points are classified as belonging to one of the plurality of classes and assigned a corresponding signal class. An

undeteriorated signal is output corresponding to the input signal in accordance with the input signal class ID. Block data may be generated from the plurality of data points.

A flow diagram of an embodiment is shown in **Figure 11**. The flow chart of **Figure 11** shows the basic processing stream for generating an undeteriorated signal, i.e., a recovered signal, from the deteriorated input signal which contains lost/damaged data. At step 1115, the preprocessing for peripheral erroneous pixels is performed. For example, an erroneous pixel may be replaced with horizontal neighboring data when there are no horizontal errors. If horizontal errors exist, the erroneous pixel may be replaced with vertical neighboring data. If vertical errors also exist, the erroneous pixel may be replaced with previous frame data.

At step 1117, each classification regarding the deteriorated input signal is executed to generate a class ID. Some class taps are selected adaptively according to another class ID. Multiple classifications may be executed, such as motion classification, error classification, spatial activity classification and spatial classification.

The classification scheme can be defined during system design, where the classification scheme, the number of classes, and other specification are decided for the target data. The design stage may include, among others, considerations of system performance and hardware complexity.

At step 1119, multiple classification generates a multiple class ID with a plurality of class IDs which are generated by various classification at step 1117. At step 1121, filter taps are adaptively selected according to the multiple class ID which is generated at step 1119. At step 1123, the filter tap structure is adaptively expanded according to the multiple class ID which is generated at step 1119. The number of filter coefficients that may be stored can be reduced

by allocating the same coefficient to multiple taps. This process is referred to as filter expansion. At step 1125, filtering with respect to the deteriorated input signal is executed to generate an undeteriorated signal. Filter coefficients are selected adaptively according to the multiple class ID which is generated in step 1119.

Ins a10 ~~For further information regarding classified adaptive error recovery, see U.S. Patent Application Serial No. ______, titled "Classified Adaptive Error Recovery Method and Apparatus", filed concurrently herewith, and is herewith incorporated by reference.~~

Although the present invention is discussed with respect to image data, the present invention may be used with any form of correlated data, including without limitation photographs or other two-dimensional static images, holograms, or other three-dimensional static images, video or other two-dimensional moving images, three-dimensional moving images, a monaural sound stream, or sound separated into a number of spatially related streams, such as stereo.

Figure 12 shows an example of an audio signal and how it is compatible with the present invention.

An example of audio signal 1201 is monitored at one or more time points t0 - t8. The level of audio signal 1201 at time points t0 - t8 is given by tap points X0 - X8. The dynamic range of the audio signal 1201 is given as the difference between the lowest level tap point X0 and the highest level tap point X4. In addition to dynamic range, the standard deviation, the Laplacian value or the spatial gradient value can be introduced for spatial activity classification.

The invention has been described in conjunction with the preferred embodiment. It is evident that numerous alternatives, modifications, variations

and uses will be apparent to those skilled in the art in light of the foregoing description.